TVSum: Summarizing Web Videos Using Titles

Yale Song, Jordi Vallmitjana, Amanda Stent, Alejandro Jaimes
Yahoo Labs, New York
{yalesong,jvallmi,stent,ajaimes}@yahoo-inc.com

Abstract

Video summarization is a challenging problem in part because knowing which part of a video is important requires prior knowledge about its main topic. We present TVSum, an unsupervised video summarization framework that uses title-based image search results to find visually important shots. We observe that a video title is often carefully chosen to be maximally descriptive of its main topic, and hence images related to the title can serve as a proxy for important visual concepts of the main topic. However, because titles are free-formed, unconstrained, and often written ambiguously, images searched using the title can contain noise (images irrelevant to video content) and variance (images of different topics). To deal with this challenge, we developed a novel co-archetypal analysis technique that learns canonical visual concepts shared between video and images, but not in either alone, by finding a joint-factorial representation of two data sets. We introduce a new benchmark dataset, TVSum50, that contains 50 videos and their shot-level importance scores annotated via crowdsourcing. Experimental results on two datasets, SumMe and TVSum50, suggest our approach produces superior quality summaries compared to several recently proposed approaches.

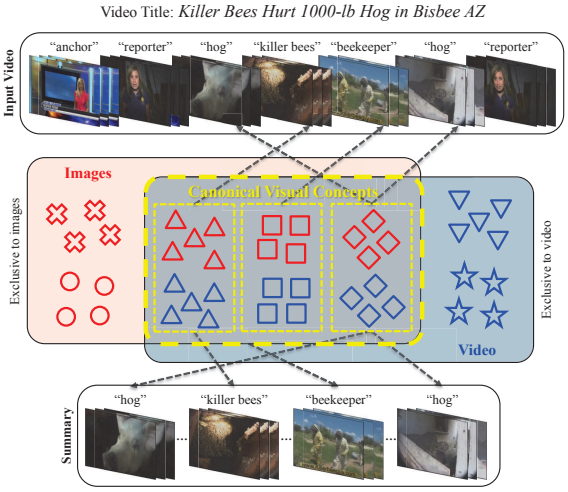

Figure 1. **An illustration of title-based video summarization.** We use title-based image search results to generate a summary by selecting shots that are the most relevant to, and representative of, canonical visual concepts shared between the given video and images. Our novel co-archetypal analysis learns canonical visual concepts by focusing on the shared region (yellow dotted rectangle area), singling out patterns that are exclusive to either set, allowing us to discard images irrelevant to video frames (and vice versa).

1. Introduction

The sheer amount of video available online has increased the demand for efficient ways to search and retrieve desired content [46]. Currently, users choose to watch a video based on various metadata, e.g., thumbnail, title, description, video length, etc. This does not, however, provide a concrete sense of the actual video content, making it difficult to find desired content quickly [48]. Video summarization aims to provide this information by generating the *gist* of a video, benefiting both the users (through a better user experience) and companies that provide video streaming and search (with increased user engagement).

Video summarization is a challenging problem in part because knowing which part of a video is important, and thus "summary worthy," requires prior knowledge about its main topic. Conventional approaches produce a summary by defining a generic scheme, e.g., based on content frequency [52] or non-redundancy [12, 50]; while simple and efficient, visual frequency is not directly related to topicality. Recent work has addressed this issue by using images and videos collected from the web, learning a predictive model tailored to a predefined set of categories [28, 38, 43] or constrained to a limited object domain [25]; while promising, it remains as a challenge to deal with the variety of topics in online videos.

We present *TVSum* (Title-based Video Summarization), an unsupervised video summarization framework that uses the video title to find visually important shots. We observe that a title is often carefully chosen to describe its main topic, and thus serves as a strong prior on the expected summary. This motivates us to collect title-based image search results, and use them to select shots that are

978-1-4673-6964-0/15/$31.00 ©2015 IEEE

the most relevant to, and representative of, "canonical visual concepts" shared between video and images (Figure 1). Previous works have explored a similar direction with web images [25, 26], but with an assumption that an input query is in the form of topical keywords, making it easier to obtain a compact set of images representing the main topic. However, this is far from the case with the titles of online videos: They are free-formed, unconstrained, and often written ambiguously. Consequently, images searched using the title can contain noise (images irrelevant to the video) and variance (images of different topics). This makes it particularly difficult to learn canonical visual concepts shared between video and images [19, 29].

To deal with this challenge, we present *co-archetypal analysis* that learns canonical visual concepts by focusing on the patterns shared between video and images; we call such patterns *co-archetypes*. Our method learns a joint-factorial representation of two data sets that are *conditionally independent* of each other, given co-archetypes. Unlike archetypal analysis [13, 11], we incorporate a regularization term that penalizes the deviation between the factorizations of video and images with respect to the co-archetypes. We develop an efficient optimization algorithm using block-coordinate descent, and demonstrate its advantage over archetypal analysis on synthetic and real data.

We introduce a new dataset, *TVSum50*, that contains 50 videos representing various genres (e.g., news, how-to's, user generated content), and their shot-level importance scores annotated via crowdsourcing. Unlike existing annotation methods [20, 38], we avoid *chronological bias* (shots are scored higher if appearing early in a video) by pre-clustering and randomization, resulting in a high degree of inter-rater reliability, i.e., Cronbach's alpha of 0.81.

In summary, we present Title-based Video Summarization (TVSum) that generates a summary by selecting shots that are the most relevant to, and representative of, canonical visual concepts shared between video and title-related images. Experimental results on two datasets, SumMe [20] and our TVSum50, show that our title-based video summarization framework significantly outperforms several baseline approaches. Our main technical contribution is in the development of a co-archetypal analysis technique that learns a joint-factorial representation of a video and images, focusing on the patterns shared only between the two sets.

2. Related Work

Some of early work in video summarization focused on videos of certain genres, such as news [36], sports [9, 45], surveillance [18, 40], and egocentric [27, 32], and generated summaries by leveraging genre-specific information, e.g., salient objects in sports [17], and salient regions in egocentric videos [27, 32]. However, they are difficult to apply to *online video* summarization because such genre-specific assumptions generally do not hold for web-scale videos.

Much work has been devoted to unsupervised video summarization, assessing the importance of frames using visual attention [16, 5], interestingness [21, 20], user engagement [33, 35], content frequency [52] and non-redundancy [30, 50]. More recently, summarizing videos using web images has attracted much attention [25, 26]. Based on an insight that images tend to capture objects of interest from the optimal viewpoint, Khosla *et al*. [25] learn canonical viewpoints using a multi-class SVM, while Kim *et al*. [26] use a diversity ranking model. They make an implicit assumption that descriptive topical keywords are given already (e.g., cars and trucks [25]), but user-provided keywords are often imprecise and uninformative [10, 31, 51], making them difficult to apply in the real-world setting without human supervision. Our work contributes to this line of work by showing how to use titles for summarizing videos without any human supervision, with a technique that learns canonical visual concepts shared only between video and title-based image search results.

3. TVSum Framework

Our framework consists of four modules: shot segmentation, canonical visual concept learning, shot importance scoring, and summary generation. First, we group sequences of visually coherent frames into shots so that the resulting summary contains shots rather than keyframes. Next, we learn canonical visual concepts using our novel co-archetypal analysis, which learns a joint-factorization of a video and images.[1] We then measure the importance of each frame using the learned factorial representation of the video, and combine the importance measures into shot-level scores. Finally, a summary is generated by maximizing the total importance score with a summary length budget.

Notation: $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ is a matrix of n video frames, with each column $\mathbf{x}_i \in \mathbb{R}^d$ representing a frame with a certain set of image feature descriptors. $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_m] \in \mathbb{R}^{d \times m}$ is a matrix of m images defined in a similar way. Our goal is, given $\mathbf{X}$ and $\mathbf{Y}$, to generate a summary $\mathcal{S} \in \mathbb{R}^{d \times l}$ for $l \ll n$ by concatenating a few non-overlapping and important shots given a length budget l.

3.1. Shot Segmentation

Shot segmentation [41] is a crucial step in video summarization for maintaining visual coherence within each shot, which in turn affects the overall quality of a summary. Many existing approaches use heuristics, e.g., uniform segmentation [50, 43]. Similar to [38], we cast the problem as change-point detection, a more principled statistical method to find "changing moments" in a time-series sequence [1]. While [38] uses a kernel-based approach, we instead frame the task as a group LASSO problem, based on [4].

[1] Section 4 describes how we collect web images from a video title.

Given a matrix $\mathbf{X}$ and the number of change-points k, the problem reduces to finding a piecewise-constant approximation $\mathbf{H} \in \mathbb{R}^{d \times n}$ such that $\mathbf{H}$ minimizes the reconstruction error: $\min_{\mathbf{H}} \|\mathbf{X}-\mathbf{H}\|$. Change points are then found by taking the first-order discrete derivative of $\mathbf{H}$ over column vectors and finding non-zero locations. It is well known that dynamic programming (DP) provides a globally optimal solution to this problem in $\mathcal{O}(dn^2k)$ [1]. Unfortunately, this assumes k is given, which is in general unavailable in the real-world setting. Also, the quadratic complexity makes DP impractical for videos with even a modest duration (a 5 minute video consists of about 9,000 frames); we need a more computationally efficient solution.

Formulation: In this work, we find change-points by solving a convex problem with total variation [22, 4]:

$$\min_{\mathbf{H}} \frac{1}{2}\|\mathbf{X}-\mathbf{H}\|_{\mathrm{F}}^2 + \lambda \sum_{t=1}^{n-1} \|\mathbf{H}_{\cdot,t+1} - \mathbf{H}_{\cdot,t}\|_2 \qquad (1)$$

where $\|.\|_{\mathrm{F}}$ is the Frobenius norm. The first term measures the reconstruction error, the second measures the total variation, and $\lambda > 0$ controls the relative importance between the two. Note that the total variation is penalized with a sparsity-inducing $\ell_{2,1}$ norm. This allows us to find a solution $\mathbf{H}$ that is *column-wise sparse*; without this group sparsity constraint, change-points may appear at different locations across d dimensions, making it difficult to interpret.

Optimization: We solve Equation 1 using the group LARS [49] algorithm. Given the maximum number of change points k (set to half the video duration in seconds), we compute an approximate (piecewise-linear) regularization path by iterating over k steps and adding a change-point to an active set $\mathcal{A}$ at each step, with a choice of λ that produces over-segmentation. We then perform model selection to find the optimal $k' \leq k$ by finding a subset $\mathcal{A}' \subset \mathcal{A}$ such that it no longer improves the sum-squared errors (SSE) between $\mathbf{X}$ and $\mathbf{H}$, up to a threshold θ (set to 0.1). Our method does not require knowing the optimal k a priori thanks to the model selection. Also, the group LARS is highly efficient, i.e., $\mathcal{O}(dnk)$; although the model selection takes $\mathcal{O}(k^3)$ for computing SSE for all k, in practice $k \ll n$. This makes our method faster and more practical than the DP solution.

3.2. Canonical Visual Concept Learning

We define canonical visual concepts as the patterns shared between video $\mathbf{X}$ and its title-based image search results $\mathbf{Y}$, and represent them as a set of p latent variables $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_p] \in \mathbb{R}^{d \times p}$, where $p \ll d$ (in our experiments, p=200 and d=1,640).

Motivated by archetypal analysis [13], we find $\mathbf{Z}$ by learning the factorizations of $\mathbf{X}$ and $\mathbf{Y}$ with respect to $\mathbf{Z}$ under two geometrical constraints: (i) each video frame $\mathbf{x}_i$ and image $\mathbf{y}_i$ should be well approximated by a convex combination of latent variables $\mathbf{Z}$; (ii) each latent variable $\mathbf{z}_j$ should be well approximated *jointly* by a convex combination of video frames $\mathbf{X}$ and by a convex combination of images $\mathbf{Y}$. Therefore, given $\mathbf{Z}$, each video frame $\mathbf{x}_i$ and image $\mathbf{y}_i$ is approximated by $\mathbf{Z}\boldsymbol{\alpha}_i^{\mathrm{X}}$ and $\mathbf{Z}\boldsymbol{\alpha}_i^{\mathrm{Y}}$, respectively, where $\boldsymbol{\alpha}_i^{\mathrm{X}}$ and $\boldsymbol{\alpha}_i^{\mathrm{Y}}$ are coefficient vectors in the unit simplex Δ^p:

$$\Delta^p = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^p \mid \sum_{j=1}^{p} \alpha[j] = 1 \text{ and } \alpha[j] \geq 0 \text{ for all } j \right\}$$

While $\mathbf{x}_i$ and $\mathbf{y}_i$ are approximated in terms of $\mathbf{Z}$ independently of each other, each $\mathbf{z}_j$ is *jointly* approximated by $\mathbf{X}\boldsymbol{\beta}_j^{\mathrm{X}}$ and $\mathbf{Y}\boldsymbol{\beta}_j^{\mathrm{Y}}$, i.e., $\mathbf{z}_j \approx \mathbf{X}\boldsymbol{\beta}_j^{\mathrm{X}} \approx \mathbf{Y}\boldsymbol{\beta}_j^{\mathrm{Y}}$, where $\boldsymbol{\beta}_j^{\mathrm{X}}$ and $\boldsymbol{\beta}_j^{\mathrm{Y}}$ are coefficient vectors in Δ^n and Δ^m, respectively. This joint approximation encourages $\mathbf{Z}$ to capture canonical visual concepts that appear both in a video and an image set, but not in either alone. Inspired by [13], we refer to the latent variables $\mathbf{Z}$ as *co-archetypes* of $\mathbf{X}$ and $\mathbf{Y}$, and the process of finding $\mathbf{Z}$ as *co-archetypal analysis*.

Formulation: Co-archetypal analysis is formulated as an optimization problem that finds a solution set $\boldsymbol{\Omega} = \{\mathbf{A}^{\mathrm{X}}, \mathbf{B}^{\mathrm{X}}, \mathbf{A}^{\mathrm{Y}}, \mathbf{B}^{\mathrm{Y}}\}$ by solving the following objective:

$$\min_{\boldsymbol{\Omega}} \|\mathbf{X}-\mathbf{Z}\mathbf{A}^{\mathrm{X}}\|_{\mathrm{F}}^2 + \|\mathbf{Y}-\mathbf{Z}\mathbf{A}^{\mathrm{Y}}\|_{\mathrm{F}}^2 + \gamma\|\mathbf{X}\mathbf{B}^{\mathrm{X}}-\mathbf{Y}\mathbf{B}^{\mathrm{Y}}\|_{\mathrm{F}}^2 \quad (2)$$

where $\mathbf{A}^{\mathrm{X}} = \left[\boldsymbol{\alpha}_1^{\mathrm{X}}, \cdots, \boldsymbol{\alpha}_n^{\mathrm{X}}\right] \in \mathbb{R}^{p \times n}$, $\mathbf{B}^{\mathrm{X}} = \left[\boldsymbol{\beta}_1^{\mathrm{X}}, \cdots, \boldsymbol{\beta}_p^{\mathrm{X}}\right] \in \mathbb{R}^{n \times p}$, and similarly $\mathbf{A}^{\mathrm{Y}} \in \mathbb{R}^{p \times m}$, $\mathbf{B}^{\mathrm{Y}} \in \mathbb{R}^{m \times p}$. Given $\mathbf{Z}$, $\mathbf{X}$ and $\mathbf{Y}$ are factorized using $\mathbf{Z}$ independently of each other; this is expressed in the first and the second terms. On the other hand, $\mathbf{Z}$ is jointly factorized using $\mathbf{X}$ and $\mathbf{Y}$; this is expressed in the third term, where we penalize the deviation between two factorizations $\mathbf{X}\mathbf{B}^{\mathrm{X}}$ and $\mathbf{Y}\mathbf{B}^{\mathrm{Y}}$. The free parameter γ controls to what extend we enforce this constraint. In this work, we set $\gamma = 1.0$.

Optimization: The optimization problem in Equation 2 is non-convex, but is convex when all but one of the variables among $\boldsymbol{\Omega}$ are fixed. Block-coordinate descent (BCD) is a popular approach to solving such problems [11, 50], for its simplicity and efficiency, and for the fact that it is asymptotically guaranteed to converge to a stationary point [47]. Algorithm 1 shows our optimization procedure.

We cycle through block variables in a deterministic order ($\mathbf{A}^{\mathrm{X}}$, $\mathbf{A}^{\mathrm{Y}}$, $\mathbf{B}^{\mathrm{X}}$, then $\mathbf{B}^{\mathrm{Y}}$). For $\mathbf{A}^{\mathrm{X}}$, we solve a quadratic program (QP) on each column vector $\boldsymbol{\alpha}_i^{\mathrm{X}}$ using the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [2]:

$$\min_{\boldsymbol{\alpha} \in \Delta^p} \|\mathbf{x}_i - \mathbf{Z}\boldsymbol{\alpha}\|_2^2 \qquad (3)$$

We use the same QP solver to obtain the solution $\mathbf{A}^{\mathrm{Y}}$; these updates are expressed in Lines 6 and 9 of Algorithm 1. Solving for $\mathbf{B}^{\mathrm{X}}$ and $\mathbf{B}^{\mathrm{Y}}$ is a bit more involved because of the third term in Equation 2. When all but one column vector $\boldsymbol{\beta}_j^{\mathrm{X}}$ is fixed, the update rule for $\mathbf{B}^{\mathrm{X}}$ can be expressed as:

$$\min_{\boldsymbol{\beta} \in \Delta^n} \|\mathbf{X}-\mathbf{X}\mathbf{B}_{\mathrm{old}}^{\mathrm{X}}\mathbf{A}+\mathbf{X}(\boldsymbol{\beta}_{\mathrm{old}}^{\mathrm{X}}-\boldsymbol{\beta})\mathbf{A}_j^{\mathrm{X}}\|_{\mathrm{F}}^2 + \lambda\|\mathbf{Y}\boldsymbol{\beta}_j^{\mathrm{Y}}-\mathbf{X}\boldsymbol{\beta}\|_2^2$$

Algorithm 1 Solving an optimization of Equation 2

1: **Input**: $\mathbf{X} \in \mathbb{R}^{d\times n}$, $\mathbf{Y} \in \mathbb{R}^{d\times m}$, p, T
2: Initialize $\mathbf{Z} \in \mathbb{R}^{d\times p}$ with random columns of $\mathbf{X}$ and $\mathbf{Y}$
3: Initialize $\mathbf{B}^{\mathrm{X}}$ s.t. $\mathbf{Z} = \mathbf{X}\mathbf{B}^{\mathrm{X}}$, $\mathbf{B}^{\mathrm{Y}}$ s.t. $\mathbf{Z} = \mathbf{Y}\mathbf{B}^{\mathrm{Y}}$
4: **for** $t = 1 \cdots T$ **do**
5: **for** $i = 1 \cdots n$ **do**
6: $\boldsymbol{\alpha}_i^{\mathrm{X}} \leftarrow \arg\min_{\boldsymbol{\alpha}} \|\mathbf{x}_i - \mathbf{Z}\boldsymbol{\alpha}\|_2^2$
7: **end for**
8: **for** $i = 1 \cdots m$ **do**
9: $\boldsymbol{\alpha}_i^{\mathrm{Y}} \leftarrow \arg\min_{\boldsymbol{\alpha}} \|\mathbf{y}_i - \mathbf{Z}\boldsymbol{\alpha}\|_2^2$
10: **end for**
11: $\mathbf{R}^{\mathrm{X}} \leftarrow \mathbf{X} - \mathbf{Z}\mathbf{A}^{\mathrm{X}}$, $\mathbf{R}^{\mathrm{Y}} \leftarrow \mathbf{X} - \mathbf{Z}\mathbf{A}^{\mathrm{Y}}$
12: **for** $j = 1 \cdots p$ **do**
13: $\boldsymbol{\beta}_j^{\mathrm{X}} \leftarrow \arg\min_{\boldsymbol{\beta}} \|\mathbf{R}^{\mathrm{X}} + (\mathbf{z}_j - \mathbf{X}\boldsymbol{\beta})\mathbf{A}_j^{\mathrm{X}}\|_{\mathrm{F}}^2 + \lambda\|\mathbf{z}_j - \mathbf{X}\boldsymbol{\beta}\|_2^2$
14: $\boldsymbol{\beta}_j^{\mathrm{Y}} \leftarrow \arg\min_{\boldsymbol{\beta}} \|\mathbf{R}^{\mathrm{Y}} + (\mathbf{z}_j - \mathbf{Y}\boldsymbol{\beta})\mathbf{A}_j^{\mathrm{Y}}\|_{\mathrm{F}}^2 + \lambda\|\mathbf{z}_j - \mathbf{Y}\boldsymbol{\beta}\|_2^2$
15: $\mathbf{R}^{\mathrm{X}} \leftarrow \mathbf{R}^{\mathrm{X}} + (\mathbf{z}_j - \mathbf{X}\boldsymbol{\beta}_j^{\mathrm{X}})\mathbf{A}_j^{\mathrm{X}}$
16: $\mathbf{R}^{\mathrm{Y}} \leftarrow \mathbf{R}^{\mathrm{Y}} + (\mathbf{z}_j - \mathbf{Y}\boldsymbol{\beta}_j^{\mathrm{Y}})\mathbf{A}_j^{\mathrm{Y}}$
17: $\mathbf{z}_j \leftarrow \frac{1}{2}\left(\mathbf{X}\boldsymbol{\beta}_j^{\mathrm{X}} + \mathbf{Y}\boldsymbol{\beta}_j^{\mathrm{Y}}\right)$
18: **end for**
19: **end for**
20: Return $\mathbf{A}^{\mathrm{X}}, \mathbf{B}^{\mathrm{X}}, \mathbf{A}^{\mathrm{Y}}, \mathbf{B}^{\mathrm{Y}}$

where $\mathbf{A}_j^{\mathrm{X}}$ is the j-th row vector. Letting $\mathbf{R}^{\mathrm{X}} = \mathbf{X} - \mathbf{Z}\mathbf{A}^{\mathrm{X}}$, we can rewrite the above formular as

$$\min_{\boldsymbol{\beta}\in\Delta^n} \|\mathbf{R}^{\mathrm{X}} + (\mathbf{z}_j - \mathbf{X}\boldsymbol{\beta})\mathbf{A}_j^{\mathrm{X}}\|_{\mathrm{F}}^2 + \lambda\|\mathbf{z}_j - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad (4)$$

We use a generic QP solver to solve Equation 4 and similarly for $\mathbf{B}^{\mathrm{Y}}$; these updates are expressed in Lines 13 and 14 of Algorithm 1. We acknowledge that we can solve this more efficiently by implementing a customized QP solver using an SMO-style algorithm [37], expressing the unit simplex condition as constraints and solving the problem blockwise (two variables at a time); we leave this as future work.

3.3. Shot Importance Scoring

We first measure frame-level importance using the learned factorization of $\mathbf{X}$ into $\mathbf{XBA}$. Specifically, we measure the importance of the i-th video frame $\mathbf{x}_i$ by computing the total contribution of the corresponding elements of $\mathbf{BA}$ in reconstructing the original signal $\mathbf{X}$, that is,

$$\texttt{score}(\mathbf{x}_i) = \sum_{j=1}^{n} \mathbf{B}_i\boldsymbol{\alpha}_j \quad (5)$$

where $\mathbf{B}_i$ is the i-th row of the matrix $\mathbf{B}$. Recall that each $\mathbf{x}$ is a convex combination of co-archetypes $\mathbf{Z}$, and each $\mathbf{z}$ is a convex combination of data $\mathbf{X}$ (and of $\mathbf{Y}$). Intuitively, this "chain reaction"-like formulation makes our scoring function measure how representative a particular video frame $\mathbf{x}_i$ is in the reconstruction of the original video $\mathbf{X}$ using $\mathbf{Z}$. Also, the joint-formulation of $\mathbf{Z}$ using $\mathbf{X}$ and $\mathbf{Y}$ allows us to measure the relevance of $\mathbf{x}_i$ to the canonical visual concepts $\mathbf{Z}$ shared between the given video and images. We then compute shot-level importance scores by taking an average of the frame importance scores within each shot.

Consequently, our definition of importance captures the relevance and the representativeness of each shot to the canonical visual concepts. Note that, by incorporating the notion of representativeness, it assigns low scores to similar looking, yet less representative frames; hence, it implicitly removes redundant information in the summary.

3.4. Summary Generation

To generate a summary of length l, we solve the following optimization problem:

$$\max \sum_{i=1}^{s} u_i v_i \text{ subject to } \sum_{i=1}^{s} u_i w_i \leq l,\ u_i \in \{0,1\} \quad (6)$$

where s is the number of shots, v_i is the importance score of the i-th shot, and w_i is the length of the i-th shot. Note that this is exactly the 0/1 knapsack problem; we solve it using dynamic programming. The summary is then created by concatenating shots with $u_i \neq 0$ in chronological order. Following [20], we set l to be 15% of the video length.

4. TVSum50 Benchmark Dataset

Title-based video summarization is a relatively unexplored domain; there is no publicly available dataset suitable for our purpose. We therefore collected a new dataset, *TVSum50*, that contains 50 videos and their shot-level importance scores obtained via crowdsourcing.

4.1. Video Data Collection

We selected 10 categories from the TRECVid Multimedia Event Detection (MED) task [42] and collected 50 videos (5 per category) from YouTube using the category name as a search query term. From the search results, we chose videos using the following criteria: (i) under the Creative Commons license; (ii) duration is 2 to 10 minutes; (iii) contains more than a single shot; (iv) its title is descriptive of the visual topic in the video. We collected videos representing various genres, including news, how-to's, documentaries, and user-generated content (vlog, egocentric). Figure 2 shows thumbnails of the 50 videos and their corresponding categories; Table 1 shows descriptive statistics; we provide a full list of the 50 videos with their titles and genres in supplementary material.

4.2. Web Image Data Collection

In order to learn canonical visual concepts from a video title, we need a sufficiently diverse set of images [15]. Unfortunately, the title itself can sometimes be too specific as a query term [29]. Here, we perform query expansion of the title using a simple method. We initialize a query set $\mathcal{Q}$ with the video title. We then tokenize the title using common delimiters (i.e., '.', '-', '=', '—', ';', ':') and remove

Figure 2. **TVSum50 dataset** contains 50 videos collected from YouTube using 10 categories from the TRECVid MED task [42] as search queries: changing Vehicle Tire (VT), getting Vehicle Unstuck (VU), Grooming an Animal (GA), Making Sandwich (MS), ParKour (PK), PaRade (PR), Flash Mob gathering (FM), BeeKeeping (BK), attempting Bike Tricks (BT), and Dog Show (DS).

Cat.	Descriptive Statatistics			Label Consistency	
	#vid	#frm	length	Pair. F_1	Cron. α
VT	5	39,841	25m25s	.38 (.06)	.88 (.04)
VU	5	35,014	19m28s	.36 (.07)	.78 (.13)
GA	5	30,920	18m7s	.36 (.04)	.87 (.03)
MS	5	37,095	24m58s	.37 (.05)	.83 (.06)
PK	5	41,634	24m50s	.34 (.06)	.74 (.09)
PR	5	44,042	25m3s	.34 (.03)	.82 (.06)
FM	5	30,747	18m37s	.34 (.04)	.79 (.07)
BK	5	30,489	17m30s	.34 (.07)	.80 (.10)
BT	5	25,747	14m39s	.41 (.06)	.87 (.03)
DS	5	36,827	20m59s	.33 (.04)	.76 (.08)
Total	50	352,356	3h29m42s	.36 (.05)	.81 (.08)

Table 1. **Descriptive statistics and human labeling consistency of our TVSum50 dataset**. For human label consistency, we report means and standard deviations of a pairwise F_1 score and Cronbach's alpha. Our labeling results show a higher degree of consistency than the SumMe dataset [20], which reports an average pairwise F_1 score of 0.31 and Cronbach's alpha of 0.74.

stop words and special characters; the resulting tokens are added to $\mathcal{Q}$. Lastly, we create additional queries by taking n-grams around each token [8]; we set $n = 3$. Once the query set $\mathcal{Q}$ is built, we collect up to 200 images per query using Yahoo! image search engine.

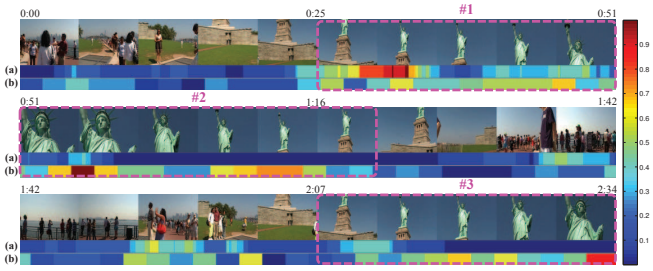

Figure 3. **Chronological bias in shot importance labeling.** Shown here is the video "Statue of Liberty" from the SumMe dataset [20]. Segments #1-3 (dotted boxes) contain the highlights, i.e., the statue. Color bars in each row represent normalized shot importance scores: (a) from the SumMe dataset, (b) collected by us using our annotation interface. Scores in (a) show the effect of chronological bias, where segments #2 and #3 get lower scores than #1, because they appear later in the video and users evaluated shots chronologically. Our annotation interface avoids this problem, resulting in consistent scores for visually similar shots.

4.3. Crowd-sourced Annotation

Due to the subjectivity inherent in the task, it is infeasible to obtain clear-cut ground truth labels in video summarization; thus, evaluation is often carried out using human judgments. One popular approach involves asking humans to watch several versions of summaries and to assess the quality of each in comparison to the others, e.g., by voting for the best summary [27, 32]. While simple and fast, this approach does not scale well because the study has to be re-run every time a change is made. Another approach involves asking humans to watch the whole video (instead of just summaries) and to assess the importance of every part of the video; the responses are then treated as gold standard labels [25, 20, 38]. This approach has the advantage that, once the labels are obtained, experiments can be carried out indefinitely, which is desirable especially for computer vision systems that involve multiple iterations and testing. We take the latter approach in this work.

Task setup: We used Amazon Mechanical Turk to collect 1,000 responses (20 per video). During the task, a participant was asked to (1) read the title first (to simulate a typical online video browsing scenario); (ii) watch the whole video in a single take; (iii) provide an importance score of 1 (not important) to 5 (very important) to each of uniform-length shots for the whole video. We empirically found that a shot length of two seconds is appropriate for capturing local context with good visual coherence. We muted audio to ensure that scores are based solely on visual stimuli.

Avoiding chronological bias: We observed that humans have a tendency to assign higher scores to the shots that appear earlier in video, simply by virtue of their temporal precedence, regardless of their actual visual quality or rep-

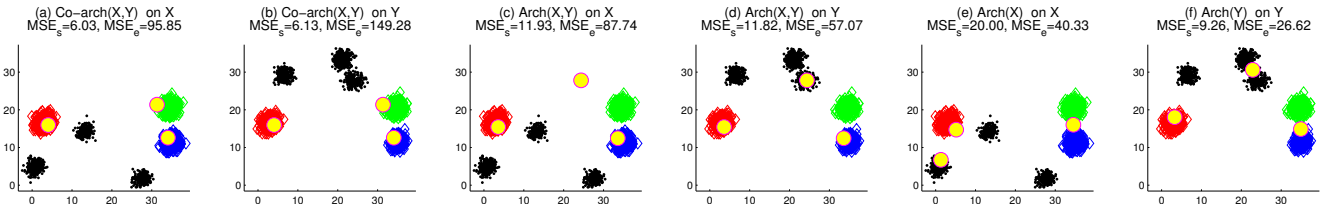

Figure 4. **Empirical analysis of co-archetypal analysis on synthetic data.** We show scatter plots of samples from each data set: (a,c,e) $\mathbf{X}$, (b,d,f) $\mathbf{Y}$. Samples from clusters shared between $\mathbf{X}$ and $\mathbf{Y}$ are shown in color (red, green, blue); samples from clusters exclusive to each dataset are shown in black. Yellow dots represent learned co-archetypes $\mathbf{Z}$. We show results of three different approaches: (a-b) our co-archetypal analysis; (c-d) archetypal analysis [11] using a combined dataset $[\mathbf{X}; \mathbf{Y}]$; (e-f) archetypal analysis using each dataset independently. As seen in (a-b), our approach learns $\mathbf{Z}$ that is more aligned with the locations of shared cluster centroids. We also show mean squared errors $\frac{1}{n}\sum_i \|\mathbf{x}_i - \mathbf{Z}\boldsymbol{\alpha}_i\|_2^2$ of samples from shared clusters (MSE$_s$) and from exclusive clusters (MSE$_e$); ideally, we want low MSE$_s$ and high MSE$_e$. Our approach (a-b) achieves the minimum MSE$_s$ with maximum MSE$_e$.

resentativeness (see Figure 3); we call this phenomenon a *chronological bias* [3]. This is problematic because such labels are biased toward summarization methods that directly minimize content redundancy over time (e.g., [50]), which, obviously, is not the optimal solution for videos whose visual content is mostly similar but differs in quality and representativeness, e.g., landmark videos. To address this issue, we pre-clustered shots using k-means (k set to the video length, in seconds, divided by 10) and presented the shots within each cluster in random order. We found empirically that our approach provides more consistent and meaningful scores than the chronological evaluation scheme (compare Figure 3 (a) and (b)). Note that preserving the correct temporal ordering can be important in certain cases [32, 39, 44]; we provided an option to re-watch the original video, if needed, at any time during the task so that the participant can figure out where each shot appears in a video.

Regularizing score distributions: We need to ensure that the score distribution is appropriate for generating summaries. Ideally, the distribution is skewed toward low scores so that only a few shots get high scores. To this end, we defined target ranges for each score assignment that participants must follow: the relative frequency of shots assigned to score 5 should be between 1% and 5%; score 4 should be between 5% to 10%; score 3 should be between 10% and 20%; score 4 should be between 20% and 40%; and score 5 gets the rest. This allows us to regularize score distributions across participants and videos.

4.4. Evaluation Metric

Motived by [23] and similar to [20], we assess the quality of an automatically generated summary by measuring agreement between the summary and gold standard labels provided by the crowd. Specifically, we compute an average pairwise F$_\beta$-measure. Given a proposed summary $\mathcal{S}$ and a set of n gold-standard summaries $\mathcal{G}$, we compute the precision p_i and the recall r_i for each pair of $\mathcal{S}$ and $\mathcal{G}_i$. The average pairwise F$_\beta$-measure is then computed as

$$\tilde{F}_\beta = \frac{1}{N}\sum_{i=1}^{n} \frac{(1+\beta^2) \times p_i \times r_i}{(\beta^2 \times p_i) + r_i} \tag{7}$$

where β balances the relative importance between precision and recall; we set $\beta = 1$. Table 1 includes the average pairwise F$_1$-measure computed among the gold standard labels.

5. Experiments

5.1. Co-archetypal Analysis on Synthetic Data

Co-archetypal analysis aims to learn $\mathbf{Z}$ that captures canonical patterns shared between two datasets (here, video and images). To demonstrate the effectiveness of our method in an objective manner, we performed an experiment on synthetic data, comparing our approach to archetypal analysis [11].

Dataset: For each dataset $\mathbf{X}$ and $\mathbf{Y}$, we defined six cluster centroids in 2-dimensional space at random, where three of the six are shared between two sets. Then, for each cluster, we drew 200 random vectors from a bivariate Normal distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where $\boldsymbol{\mu}_i \in \mathbb{R}^2$ is set to the i-th cluster centroid and $\boldsymbol{\Sigma}_i \in \mathbb{R}^{2\times 2}$ is set to be a symmetric positive semi-definite random matrix. Because there are three shared clusters, we set the number of co-archetypes $p = 3$ in all three test conditions.

Result: As seen in Figure 4, our approach (a-b) learns $\mathbf{Z}$ that is more aligned with the true locations of shared cluster centroids. Ideally, we want the reconstruction error to be lower for samples from shared clusters, and higher for samples from exclusive clusters. This represents our scenario where video frames and images share canonical visual concepts, and where we want to summarize videos using frames that are shared between the two sets. Our results indicate that only co-archetypal analysis has this desirable property, achieving the minimum MSE$_s$ and the maximum MSE$_e$.

5.2. Video Summarization on Real-world Data

Datasets: We evaluated our approach on two real-world datasets: SumMe [20] and our TVSum50. Some video titles in the SumMe dataset are abbreviated to generic terms, making them less descriptive, e.g., "Jumps"; we substituted those titles with either their original or more descriptive titles.[2] We collected web images for both datasets using the procedure described in Section 4.2.

Features: We extract a set of standard image descriptors: RGB/HSV color histograms to capture global color activity, a pyramid of HoG (pHoG) [7] to capture local and global shape information, GIST [34] to capture global scene information, and dense SIFT (dSIFT) [6] to capture local appearance information. The color histograms are computed on both RGB and HSV images using 32 bins, producing a 192D vector; all other descriptors are computed on a gray scale image. The pHoG descriptor is computed over a 4-level pyramid using 8 bins, producing a 680D vector. The GIST descriptor is computed using 4 x 4 blocks and 8 orientations per scale, producing a 512D vector. The dSIFT descriptor is computed using a bin size of 9 and step size of 8, and represented as bag-of-words by learning a codebook of size 256 using k-means on a 5% subset of the descriptors uniformly sampled from the video frames. We concatenate the four descriptors to obtain a 1640D feature vector.

5.2.1 Baseline Models

We compared our summarization approach to 8 baselines covering a variety of different methods, including ones that leverage information from images (WP and AA_2). We used the same shot boundary information across all approaches.

Sampling (SU and SR): A summary is generated by selecting shots either uniformly (SU) or randomly (SR) such that the summary length is within the length budget l.

Clustering (CK and CS): We perform clustering on video frames and compute the distance of each frame to the closest centroid. We then compute the shot-level distances by taking an average distance of only those frames that belong to the most frequently occurring cluster within each shot. A summary is generated by selecting the most representative shots from each cluster, each of which is the closest to the centroid of the top k' largest clusters, with length budget l. We tested two clustering approaches: k-means (CK) and spectral clustering (CS), with k set to 100, following [25].

LiveLight [50] (LL): A summary is generated by removing redundant shots over time, measuring redundancy using a dictionary of shots updated online, and including a shot in the summary if the reconstruction error is larger than a threshold ϵ_0 ($\epsilon_0 = 1.0$). This approach is not able to control the summary length a priori; we selected shots with the highest reconstruction errors that fit in the length budget l.

[2]We provide a list of substituted titles in supplementary material.

Cat	SU	SR	CK	CS	LL	WP	AA_1	AA_2	CA
VT	0.39	0.29	0.33	0.39	0.47	0.36	0.33	0.38	**0.52**
VU	0.43	0.31	0.40	0.37	0.52	0.48	0.36	0.36	**0.55**
GA	0.32	0.36	0.37	0.39	**0.46**	0.35	0.28	0.33	0.41
MS	0.37	0.32	0.34	0.39	0.45	0.40	0.36	0.32	**0.58**
PK	0.36	0.32	0.34	0.39	**0.49**	0.27	0.35	0.39	0.44
PR	0.38	0.30	0.34	0.38	0.42	0.36	0.37	0.42	**0.53**
FM	0.32	0.30	0.33	0.37	0.42	0.39	0.41	0.38	**0.51**
BK	0.34	0.32	0.34	0.38	0.44	0.41	0.28	0.28	**0.47**
BT	0.36	0.29	0.35	0.46	0.45	0.32	0.24	0.27	**0.49**
DS	0.33	0.34	0.34	0.38	**0.52**	0.31	0.32	0.35	0.48
Avg	0.36	0.32	0.35	0.39	0.46	0.36	0.33	0.35	**0.50**

Table 2. **Experimental results on our TVSum50 dataset**. Numbers show mean pairwise F_1 scores. In each row, the best performing score is bold-faced; the second best is underlined. Overall, our approach (CA) statistically significantly outperforms all baselines ($p < .01$), except for LL [50] ($p = .05$).

Web Image Prior [25] (WP): As in [25], we defined 100 positive and 1 negative classes, using images from other videos in the same dataset as negative examples. This approach produces keyframes as a summary; we used a similar approach to the clustering-based method (above) to generate our summary with shots, computing the shot-level distances and selecting those shots that are most representative of the top k' largest clusters, and with length budget l.

Archetypal Analysis [11] (AA_1 and AA_2): We include two versions of archetypal analysis: one that learns archetypes from video data only (AA_1), and another that uses a combination of video and image data (AA_2). In both cases, and also for our co-archetypal analysis (CA), we set the number of archetypes $p = 200$, following [50].

5.2.2 Results and Discussion

Table 2 shows that our approach (CA) statistically significantly outperformed all other baselines ($p < .01$) on the TVSum50 dataset, except for LiveLight ($p = .05$). Figure 5 also shows our approach achieved the highest overall score of 0.2655 on the SumMe dataset (the previous state-of-the-art was of 0.234 in [20]). The performance differences were statistically significant with Sampling, Clustering, and Attention [16] ($p < .01$). We believe the closer performance gap between ours and LiveLight on the SumMe dataset is attributed to chronological bias. Indeed, Livelight achieved the highest score of 0.258 on *Statue of Liberty* by including shots with temporal precedence (see Figure 3).

Our approach performed particularly well on videos that have their visual concepts described well by images, e.g., *St. Maarten Landing* contains a famous low-altitude flyover landing approach at Mano Beach, and the images depict this concept well. Figure 6 shows that the importance scores estimated by our approach matches closely with those of humans, and that the learned co-archetypes have successfully found patterns that are shared between video and images.

Importance of joint factorization: Both AA_2 and our

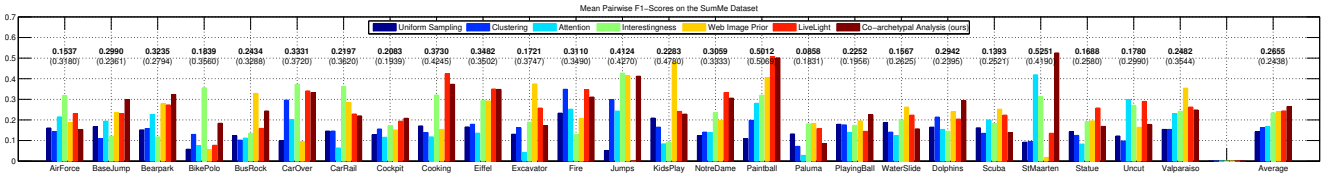

Figure 5. **Experimental results on the SumMe dataset [20]**. We include results reported in [20] for comparison with the state-of-the-art (uniform sampling, clustering, visual attention [16], and visual interestingness [20]). Bold-faced numbers represent our results; numbers in parentheses represent the highest score from any baseline. The last column shows average scores. Overall, our approach achieves the best performance (mean pairwise F_1 score of 0.2655); Interestingness achieved 0.2345, Web Image Prior achieved 0.2403, and LiveLight achieved 0.2438. Differences with three baselines (sampling, clustering, attention) were statistically significant ($p < 0.01$)

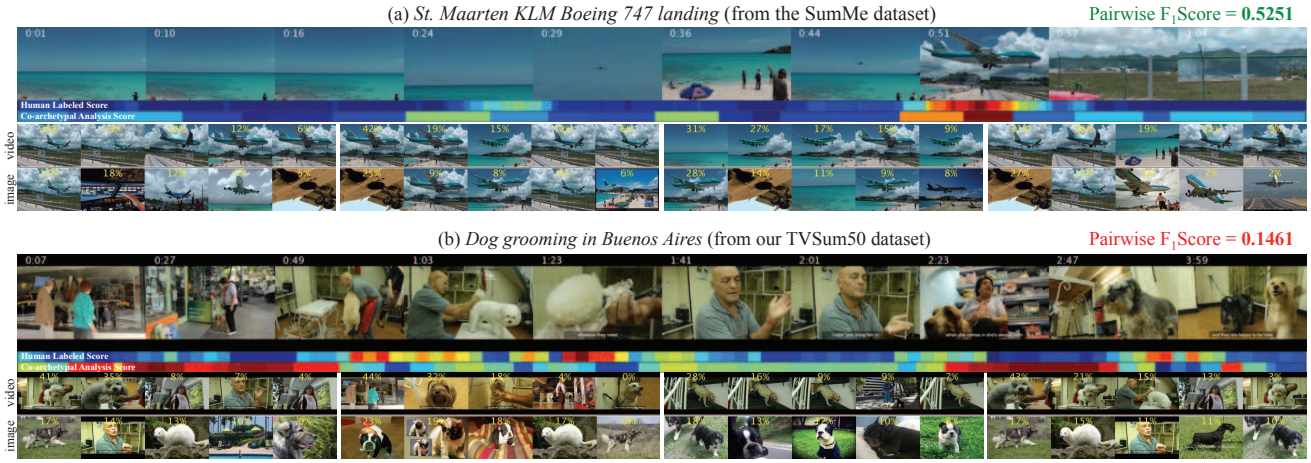

Figure 6. **Detailed results of co-archetypal analysis**. Color bars show heatmap representations of the normalized importance scores (one from human labels, another from our approach); small images show the decomposition of co-archetypes in terms of video and images (numbers indicate the coefficients $\boldsymbol{\beta}^{\mathrm{X}}$ (video) and $\boldsymbol{\beta}^{\mathrm{Y}}$ (image)). The result (a) with a high F_1 score tend to show co-archetypes that match closely with video segments scored higher by humans (an airplane). While the result (b) had a relatively low F_1 score, most of the learned co-archetypes still captured important visual concepts induced by the title (dogs).

CA were given the same input features (video and image); the only difference was that our approach has explicitly encouraged the co-archetypes to be shared between two data sources. As can be seen in Table 2 and Figure 5, our approach significantly outperformed AA_2, suggesting the importance of performing joint factorization to learn canonical visual concepts shared between video and images.

Effectiveness of our importance scoring: We compared our importance scoring function with an approach similar to LiveLight, i.e., the reconstruction error $\|\mathbf{x}_i - \mathbf{Z}\boldsymbol{\alpha}_i^{\mathrm{x}}\|_2^2$, and found that the latter produces inferior results, with an overall score of 0.255 on the SumMe dataset and of 0.418 on our TVSum50 dataset. This demonstrates the benefit of measuring importance by the total contribution of the coefficients in the factorized representation (Equation 5).

What is learned in co-archetypes: Figure 6 visualizes co-archetypes learned from the video and image data. Although there exists some variance, the learned co-archetypes tend to show important visual concepts between video and images (here, we show examples for airplanes and dog grooming), even with the existence of images and video frames irrelevant to the main topic.

6. Conclusion

We presented *TVSum*, an unsupervised video summarization framework that uses the video title to find visually important shots. Motivated by the observation that a title is often carefully chosen to describe its main topic, we developed a framework that uses title-based image search results to select shots that are the most relevant to, and representative of, canonical visual concepts shared between the video and images. While titles are descriptive, images searched using them can contain noise (images irrelevant to video content) and variance (images of different topics). To deal with this, we developed a novel co-archetypal analysis that learns the canonical patterns shared only between two sets of data. We demonstrated the effectiveness of our framework on two datasets, SumMe and our TVSum50, significantly outperforming several baselines. Moving forward, we plan to improve our image collection procedure (e.g., [29]) and utilize other types of metadata (e.g., description, comments) for summarizing online videos. Also, we are interested in applying our co-archetypal analysis in other tasks, e.g., visual concept learning [14, 24].

References

[1] M. Basseville, I. V. Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs, 1993. 2, 3

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1), 2009. 3

[3] V. Berger. *Selection bias and covariate imbalances in randomized clinical trials*, volume 66. John Wiley & Sons, 2007. 6

[4] K. Bleakley and J.-P. Vert. The group fused lasso for multiple change-point detection. *arXiv preprint arXiv:1106.4199*, 2011. 2, 3

[5] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *PAMI*, 35(1), 2013. 2

[6] A. Bosch, A. Zisserman, and X. Muñoz. Image classification using random forests and ferns. In *ICCV*, 2007. 7

[7] A. Bosch, A. Zisserman, and X. Muñoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007. 7

[8] C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM CSUR*, 44(1), 2012. 5

[9] F. Chen and C. De Vleeschouwer. Formulating team-sport video summarization as a resource allocation problem. *CSVT*, 21. 2

[10] J. Chen, Y. Cui, G. Ye, D. Liu, and S.-F. Chang. Event-driven semantic concept discovery by exploiting weakly tagged internet images. In *ICMR*, page 1, 2014. 2

[11] Y. Chen, J. Mairal, and Z. Harchaoui. Fast and robust archetypal analysis for representation learning. In *CVPR*, 2014. 2, 3, 6, 7

[12] Y. Cong, J. Yuan, and J. Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Multimedia*, 14(1), 2012. 1

[13] A. Cutler and L. Breiman. Archetypal analysis. *Technometrics*, 36(4), 1994. 2, 3

[14] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014. 8

[15] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *CVPR*, 2012. 4

[16] N. Ejaz, I. Mehmood, and S. Wook Baik. Efficient visual attention based framework for extracting key frames from videos. *Signal Processing: Image Communication*, 28(1), 2013. 2, 7, 8

[17] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *Image Processing, IEEE Transactions on*, 12(7), 2003. 2

[18] S. Feng, Z. Lei, D. Yi, and S. Z. Li. Online content-aware video condensation. In *CVPR*, 2012. 2

[19] S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth a thousand pixels. In *CVPR*, 2013. 2

[20] M. Gygli, H. Grabner, H. Riemenschneider, and L. V. Gool. Creating summaries from user videos. In *ECCV*, 2014. 2, 4, 5, 6, 7, 8

[21] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, and L. V. Gool. The interestingness of images. In *ICCV*, 2013. 2

[22] Z. Harchaoui and C. Lévy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492), 2010. 3

[23] G. Hripcsak and A. S. Rothschild. Agreement, the f-measure, and reliability in information retrieval. *JAMIA*, 12(3), 2005. 6

[24] Y. Jia, J. T. Abbott, J. Austerweil, T. Griffiths, and T. Darrell. Visual concept learning: Combining machine vision and bayesian generalization on concept hierarchies. In *NIPS*, 2013. 8

[25] A. Khosla, R. Hamid, C. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, 2013. 1, 2, 5, 7

[26] G. Kim, L. Sigal, and E. P. Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014. 2

[27] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012. 2, 5

[28] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu. Video summarization via transferrable structured learning. In *WWW*, 2011. 1

[29] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *CVPR*, 2014. 2, 4, 8

[30] D. Liu, G. Hua, and T. Chen. A hierarchical visual model for video object summarization. *PAMI*, 32(12), 2010. 2

[31] X. Liu, Y. Mu, B. Lang, and S.-F. Chang. Mixed image-keyword query adaptive hashing over multilabel images. *TOMCCAP*, 10(2), 2014. 2

[32] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013. 2, 5, 6

[33] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. In *ACM MM*, 2002. 2

[34] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3), 2001. 7

[35] W.-T. Peng, W.-T. Chu, C.-H. Chang, C.-N. Chou, W.-J. Huang, W.-Y. Chang, and Y.-P. Hung. Editing by viewing: automatic home video summarization by viewing behavior analysis. *IEEE Multimedia*, 13(3), 2011. 2

[36] M. J. Pickering, L. Wong, and S. M. Rüger. ANSES: Summarisation of news video. In *Image and Video Retrieval*. Springer, 2003. 2

[37] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods*, pages 185–208, 1999. 4

[38] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, 2014. 1, 2, 5

[39] K. Prabhakar, S. Oh, P. Wang, G. D. Abowd, and J. M. Rehg. Temporal causality for the analysis of visual events. In *CVPR*, 2010. 6

[40] Y. Pritch, A. Rav-Acha, and S. Peleg. Nonchronological video synopsis and indexing. *PAMI*, 30(11), 2008. 2

[41] A. F. Smeaton, P. Over, and A. R. Doherty. Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding*, 114(4), 2010. 2

[42] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *MIR*, 2006. 4, 5

[43] M. Sun, A. Farhadi, and S. Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014. 1, 2

[44] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012. 6

[45] D. Tjondronegoro, Y.-P. P. Chen, and B. Pham. Integrating highlights for more complete sports video summarization. *IEEE Multimedia*, 11(4), 2004. 2

[46] B. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *TOMCCAP*, 3(1), 2007. 1

[47] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3), 2001. 3

[48] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua. Event driven web video summarization by tag localization and key-shot identification. *IEEE Multimedia*, 14(4), 2012. 1

[49] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1), 2006. 3

[50] B. Zhao and E. P. Xing. Quasi real-time summarization for consumer videos. In *CVPR*, 2014. 1, 2, 3, 6, 7

[51] G. Zhu, S. Yan, and Y. Ma. Image tag refinement towards low-rank, content-tag prior and error sparsity. In *ACM MM*, 2010. 2

[52] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *ICIP*, 1998. 1, 2